

Міністерство освіти і науки України  
Харківська національна академія міського господарства

**О. Г. Журавок, І. Л. Яковицький**

Лабораторний практикум  
«JavaScript. Програмування в Інтернет.»

з дисципліни  
«Обчислювальна техніка, алгоритмічні мови та  
програмне забезпечення»

(для студентів 2 курсу денної форми навчання освітньо-кваліфікаційного рівня бакалавр, галузі знань 0507 - «Електротехніка та електромеханіка» за напрямом підготовки 6.050701 – «Електротехніка та електротехнології», професійного спрямування «Світлотехніка та джерела світла», «Електротехнічні системи електроспоживання»)

Харків  
ХНАМГ  
2010

УДК 681.3.06

Лабораторний практикум «JavaScript. Програмування в Інтернет» з навчальної дисципліни «Обчислювальна техніка, алгоритмічні мови та програмне забезпечення» (для студентів 2-го курсу денної форми навчання освітньо-кваліфікаційного рівня бакалавр, галузі знань 0507 «Електротехніка та електромеханіка» за напрямом підготовки 6.050701 «Електротехніка та електротехнології», професійного спрямування «Світлотехніка та джерела світла», «Електротехнічні системи електроспоживання») / Харк. нац. акад. міськ. госп-ва; уклад.: О. Г. Журавок, І. Л. Яковицький – Х.: ХНАМГ, 2010. – 47 с.

Анотація:

*Світові інформаційні ресурси не є статичними. Динаміка властива інформації потребує активної ролі користувача, яка може бути забезпечена алгоритмічною обробкою інформації.*

*Навчальні матеріали практикуму розміщені на сайті дистанційного навчання ХНАМГ у розділі кафедри прикладної математики та інформаційних технологій за адресою: <http://www.ksame.kharkov.ua/moodle/course/view.php?id=349>.*

Рецензент: доцент кафедри прикладної математики та інформаційних технологій,  
к.т.н. Карпенко Микола Юрійович

Затверджено кафедрою прикладної математики і інформаційних технологій,  
протокол № 3 від 6 жовтня 2010 р.

## Зміст

Вступ.....	4
Ціль практикуму.....	4
Структура практикуму.....	4
Зауваження:.....	5
Технологія виконання практикуму.....	5
Опис завдань, інструментів, приклади.....	6
Пример 1. «Що ти мені скажеш?».....	8
Пример 2. «Так можна сформувати таблицю».....	9
Пример 3. «Скористаємося чужим».....	10
Пример 4. «Використання функції Prompt».....	11
Пример 5. «Використання функції Alert».....	13
Пример 6. «Використання функції Conform».....	15
Пример 7. «Простий рахунок (Арифметика)».....	16
Пример 8. А от і перший діалог "Що ти хочеш - Одержуй!".....	18
Пример 9. «Функції: корінь квадратний і зведення в ступінь».....	19
Пример 10. «Формування таблиці квадратів».....	21
Пример 11. Таблиця множення.....	22
Пример 12. Циклічні обчислення. Факторіал цілого числа N.....	23
Пример 13. «Підрахунок кількості символів у тексті».....	26
Пример 14. «Обчислення суми: Перерахуємо грошки в кишені».....	28
Пример 15. «Нехай він сам не свій, але ми однаково перерахуємо його грошки».....	29
Пример 16. «Функціональні кнопки. Власна функція».....	31
Пример 17. Робота з об'єктом Data.....	32
Індивідуальні завдання.....	33
Завдання 1. «Анкета слухача».....	33
Завдання 2. «Оформлення власної функції».....	33
Завдання 3. «Проста логіка».....	34
Завдання 4. «Обчислювальний процес, що розгалужується».....	34
Завдання 5. «Від числа до тексту й назад».....	35
Завдання 6. «Циклічні обчислення. Табулювання функцій».....	36
Завдання 7. «Циклічні обчислення. Сортювання та інше».....	36
Завдання 8. «Об'єктне програмування. Об'єкт String».....	37
Завдання 9. «Об'єктне програмування. Масиви - об'єкт Array».....	38
Завдання 10. «Лабіринт із мінотавром».....	39
Завдання 11.....	39
Задачі підвищеної складності.....	40
Завдання «Ритми людського життя».....	40
Завдання «Гра хрестики-нулики».....	42
Завдання «Подорож коня по шахівниці».....	43
Завдання «8 з ферзів на шахівниці».....	44
Завдання «8 мирних тур на шахівниці».....	45
Завдання «Поділимо золото предків чесно!».....	46
Список літератури.....	47

## Вступ.

Лабораторний практикум "Мова JavaScript - інструмент програмування в Інтернет" включає послідовний набір прикладів, які демонструють способи використання конструкцій мови JavaScript. Після розгляду й аналізу прикладів слухачам запропонований перелік індивідуальних завдань. Їхнє виконання дозволить розвинути практичні навички написання програм. Практикум доповнений задачами підвищеної складності, рішення яких дозволяє розвинути навички алгоритмічної обробки інформації.

## Ціль практикуму.

У рамках цього практикуму ми прагнемо до досягнення наступних цілей:

- розвинути в слухачів уміння аналізувати програмний код;
- навчити слухачів складати власні програми мовою програмування JavaScript;
- розвинути навички гарного стилю написання програм;
- розвинути уявлення про технології об'єктно-орієнтованого програмування.

Думки вголос:

- по-перше, ми, безумовно, не ставимо задачу досконального вивчення мови програмування у всіх деталях;
- по-друге, ми не бажали у черговий раз повторити матеріали з величезної кількості літературних й інших джерел, а представити власне бачення процесу навчання програмуванню.

## Структура практикуму.

Цей практикум складається з послідовності прикладів (завдань). Приклад відповідає одному з розділів у курсі вивчення мови програмування. Приклад включає реалізацію описаного завдання з використанням структур мови програмування. Опис прикладів зведений в таблицю. Нижче описані її поля:

- **Зміст приклада** - опис теми до якої наведений приклад або група прикладів, які демонструють спосіб використання конструкцій мови;
- **Використовувані ОПЕРАЦІЇ / ФУНКЦІЇ** - короткий опис використовуваних конструкцій і понять;

- **Демонстраційний приклад (дивися, як працює)** - включає посилання на HTML сторінки, які містять програмну реалізацію (програму) і описи (коментарі).

*Зауваження:*

Доступність програмного коду дозволяє слухачу відволіктися від повторного набору прикладу, і сконцентрувати увагу на:

- аналізі програмного коду;
- можливості його модифікації(зміни) для рішення власних завдань.

Деякі приклади забезпечені докладним коментарем програми (сценарію).

Ми вважаємо, що докладний коментар тексту програми допомагає слухачу зрозуміти:

- ідеї й методи програмування;
- стиль написання програмного коду;
- перехід від алгоритмічних міркувань до програмного коду;
- синтаксис запису програмного коду.

*Технологія виконання практикуму.*

Демонстраційний приклад може стати прообразом для виконання завдання лабораторної роботи. Для цього виконуйте наступні кроки:

- відкрийте приклад у браузері;
- уважно проаналізуйте зміст діалогів у сценарії;
- проаналізуйте зміст сторінки HTML після завершення сценарію;
- відкрийте вихідний код сторінки (див. довідкову інформацію використовуваного браузера);
- збережіть вихідний код у текстовому файлі за допомогою програми "Блокнот" з новим ім'ям у власній папці (каталозі);
- відредагуйте збережений файл відповідно до умов індивідуального завдання відповідної роботи лабораторного практикуму.

Для виконання завдань ми пропонуємо використати найпростіший текстовий редактор "Блокнот"("Notepad"). Він входить у стандартну установку операційної системи Windows. Ми пропонуємо програму "Блокнот" тому, що вона має простий і зрозумілий інтерфейс. Це дозволяє сконцентрувати увагу власне на програмуванні.

Для виконання завдань лабораторного практикуму можна використати будь-який спеціалізований програмний засіб для створення сценаріїв мови програмування. З інтерфейсом цієї програми Вам належить познайомитися самостійно.

## Опис завдань, інструментів, приклади.

№	Зміст приклада	Використовувані ОПЕРАЦІЇ / ФУНКЦІЇ	Демонстраційний приклад (дивися як працює)
1	Вивід інформації в документ (на сторінку HTML)	метод <b>write</b> об'єкта <b>document</b>	<ul style="list-style-type: none"> <li>Що ти мені скажеш?</li> <li>Так можна сформулювати таблицю.</li> </ul>
2	Ввід-вивід інформації на екран за допомогою убудованих функцій. Організація діалогу користувача й сценарію.	Убудовані функції мови <ul style="list-style-type: none"> <li><b>Alert</b></li> <li><b>Confirm</b></li> <li><b>Prompt</b></li> <li>Уявлення про значення, що повертає функція</li> </ul>	Скористаємося чужим.
3	Найпростіші арифметичні обчислення без використання математичних функцій	Функції перетворення текст - число: <b>ParseInt()</b> - перетворення тексту у ціле число. <b>parseFloat()</b> - перетворення тексту у число з плаваючою точкою Операція присвоювання Арифметичні операції: додавання, вирахування, множення, розподіл.	<ul style="list-style-type: none"> <li>Простий рахунок (Арифметика).</li> <li>А от і перший діалог "Що ти бажай-одержуй!".</li> </ul>
4	Організація обчислень. Об'єкт Математика. Використання убудованих математичних функцій. Обчислення квадратного кореня числа Побудова таблиці квадратів (аналог таблиці Брадіса)	Убудовані математичні функції: <ul style="list-style-type: none"> <li>Об'єкт <b>Math</b> (Математика), властивості й методи</li> <li>метод <b>pow-зведення</b> в ступінь</li> <li>метод <b>sqrt-обчислення</b> квадратного кореня</li> </ul>	<ul style="list-style-type: none"> <li>Тут цариця Математика.</li> <li>Квадратний корінь.</li> </ul>
5	Реалізація циклічних обчислень - це безумовно найбільш продуктивний додаток потужності обчислювальної техніки.	Оператори й операції для організації циклічних обчислень	<ul style="list-style-type: none"> <li>Формування таблиці множення.</li> <li>Формування таблиці квадратів.</li> </ul>

	<p>Цикл - інструмент керування обчислювальним процесом, при якому необхідна послідовність операцій (обчислень) виконується необхідне число раз. Приклади завдань, у яких використовують циклічні обчислення:</p> <ol style="list-style-type: none"> <li>1. обчислення суми чисел <ul style="list-style-type: none"> <li>о сума елементів арифметичної прогресії</li> <li>о сума елементів геометричної прогресії</li> </ul> </li> <li>2. обчислення добутку чисел <ul style="list-style-type: none"> <li>о обчислення факторіала цілого числа N</li> <li>о зведення числа X у ступінь Y</li> </ul> </li> <li>3. обчислення кількості елементів <ul style="list-style-type: none"> <li>о підрахунок кількості символів у тексті(об'єм файлу в байтах)</li> <li>о підрахунок кількості звертань до web-сторінки (відвідуваність)</li> </ul> </li> <li>4. Алгоритм завдання "Обчислення суми чисел, які користувач уводить за допомогою клавіатури".</li> <li>5. Посилення алгоритму "Обчислення суми ..». с контролем коректності інформації для обчислень.</li> </ol>	<ul style="list-style-type: none"> <li>• Оператор <b>for(...;...;...){ тіло циклу }</b></li> <li>• Оператор <b>do{ тіло циклу } while( умова виходу )</b></li> <li>• Оператор <b>while{( тіло циклу ) ( умова виходу )</b></li> <li>• Операція інкрименту-збільшення цілого числа на одиницю</li> <li>• Операція декрименту-зменшення цілого числа на одиницю</li> </ul>	<ul style="list-style-type: none"> <li>• Обчислення факторіала (3 форми).</li> <li>• Обчислення кількості входження символу в рядок.</li> <li>• Обчислення суми: Перерахуємо грошки в кишені.</li> <li>• Обчислення суми: Нехай він сам не свій, але ми однаково перерахуємо його грошки</li> </ul>
6	Власні функції. Оголошення, визначення, виклик.		<ul style="list-style-type: none"> <li>• Функціональні кнопки.</li> </ul>
7	Об'єктно-орієнтоване програмування. Об'єкти: властивості й методи.		<ul style="list-style-type: none"> <li>• Вітання з використанням об'єкта Data.</li> <li>• Підрахунок входження символу (слова) у текст із використанням об'єкта String.</li> </ul>

**Пример 1. «Що ти мені скажеш?»**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад: Вивід інформації в документ HTML.</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Вивід інформації в документ HTML.");</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>

    document.write("<B><I>Добрий день!</I></B><br>")

</SCRIPT>
</I></B>
<HR>
Фраза "Добрий день!" записана в документ за допомогою сценарію JavaScript
одним оператором<BR>
От він: <B>document.write("Добрий день!")</B>
<BR>Повний текст сценарію скрипта небагато побільше

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

Лабораторный практикум по языку JavaScript	
Пример: Вывод информации в документ HTML.	
<b>Добрий день!</b>	
Фраза "Добрий день!" записана в документ при помощи сценария JavaScript одним оператором. Вот он: <code>document.write("Добрий день!")</code> Полный текст сценария скрипта немного побольше	
Последнее обновление документа: 10/26/2010 00:03:10 Харьковская национальная академия городского хозяйства Кафедра Прикладной математики и Информационных технологий телефоны:  (+38 057) 707-31-45  (+38 057) 707-31-31 Жду пожеланий и замечаний, адрес для корреспонденции: <a href="mailto:jac@ksame.kharkov.ua">jac@ksame.kharkov.ua</a> <a href="mailto:igor.yakovitskiy@gmail.com">igor.yakovitskiy@gmail.com</a> С уважением, Яковицкий Игорь Леонидович	



**Пример 2. «Так можна сформувати таблицю».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад: формування таблиці</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Формування таблиці");
    document.write("<TABLE BORDER=1 CELSPACING=3 CELLPADDING=2
RULES=cols>");
    document.write("<Caption>Екзаменаційна відомість по дисципліні</Caption>");
    document.write("<tr>");
    document.write("<td>1</td>");
    document.write("<td>Прізвище Ім'я По батькові</td>");
    document.write("<td>Номер залікової книжки</td>");
    document.write("<td>Дата здачі</td>");
    document.write("<td>Підпис викладача</td>");
    document.write("</tr>");
    document.write("</TABLE>")
jac_show();
</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

**Лабораторный практикум по языку JavaScript**

**Пример: Формирование таблицы**

**Экзаменационная ведомость по дисциплине**

1	Фамилия Имя Отчество	Номер зачетной книжки	Дата сдачи	Подпись преподавателя
---	----------------------	-----------------------	------------	-----------------------

Последнее обновление документа: 10/25/2010 13:12:37

Харьковская национальная академия городского хозяйства

Кафедра Прикладной математики и Информационных технологий

телефоны: (+38 057) 707-31-45 , (+38 057) 707-31-31

Жду пожеланий и замечаний, адрес для корреспонденций:

[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)

[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)

С уважением, Яковицкий Игорь Леонидович

**Пример 3. «Скористаємося чужим».**

**Лабораторный практикум по языку JavaScript**





**Пример: Встроенные функции обеспечения диалога.**

- "Prompt(arg1,arg2)". - функция реализует диалог при вводе значений
- "Alert(arg)". - функция вывода информационного сообщения и приостанова работы
- "Confirm(arg)". - функция реализует диалог с пользователем в стиле Да/Нет

Последнее обновление документа: 10/25/2010 14:59:56

Харьковская национальная академия городского хозяйства

Кафедра Прикладной математики и Информационных технологий

телефоны:  (+38 057) 707-31-45  ,  (+38 057) 707-31-31  )

Жду пожеланий и замечаний, адрес для корреспонденций:

[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)

[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)

С уважением, Яковицкий Игорь Леонидович

**Пример 4. «Використання функції Prompt».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>JavaScript: Приклад використання функції Prompt</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>example_show("Організація діалогу
за допомогою функції Prompt");</SCRIPT>
<P>Сторінка приклад сценарію реалізації діалогу
<P>Діалог організований за допомогою убудованої функції - "Prompt"
<P>Аргументи функції:
<OL>
  <LI>рядок символів "..). або ім'я змінної - питання
  <LI>рядок символів "..). або ім'я змінної - значення за замовчуванням
</OL>
<P>Функція виводить на екран вікно. Фокус введення сконцентрований на цьому
вікні.
<P>Аргумент 1 - інформаційне повідомлення для користувача
<P>Аргумент 2 - значення в поля введення
<P>Вікно має елементи керування:
<UL>
  <LI>поле для введення інформації
  <LI>кнопку "Ok"
  <LI>кнопку "Cancel"
</UL>
<SCRIPT language=JavaScript type=text/javascript>
prompt("Уведіть Ваші Прізвище Ім'я","Бендер Остап");
</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує проміжний результат, крок 1:

## Лабораторный практикум по языку JavaScript

### Пример: Организация диалога при помощи функции Prompt

Страница пример сценария реализации диалога

Диалог организован при помощи встроенной функции - "Prompt"

Аргументы функции:

1. строка символов "..." или имя переменной - вопрос
2. строка символов "..." или имя переменной - значение

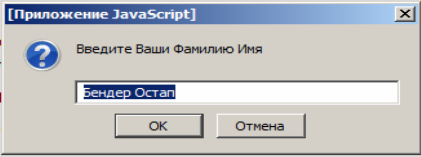
Функция выводит на экран окно. Фокус ввода сконцентрирован на поле ввода

Аргумент 1 - информационное сообщение для пользователя

Аргумент 2 - значение в поле ввода

Окно имеет элементы управления:

- поле для ввода информации
- кнопку "OK"
- кнопку "Cancel"



**Пример 5. «Використання функції Alert».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад використання функції Alert</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>example_show("Організація діалогу
за допомогою функції Alert");</SCRIPT>

<P>Сторінка приклад сценарію реалізації діалогу
<P>Діалог організований за допомогою убудованої функції - "Alert"
<P>Аргумент функції:
<OL>
  <LI>рядок символів "...". або ім'я змінної - текстове повідомлення
</OL>
<P>Функція виводить на екран вікно. Фокус введення сконцентрований на цьому
вікні.
<P>Аргумент 1 - інформаційне повідомлення для користувача
<P>Вікно має елементи керування:
<UL>
  <LI>кнопку "Ok"
</UL>
<SCRIPT language=JavaScript type=text/javascript>
  alert("Добрий ранок Країна!");
</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує проміжний результат, крок 1:

## Лабораторный практикум по языку JavaScript

### Пример: Организация диалога при помощи функции Alert

Страница пример сценария реализации диалога

Диалог организован при помощи встроенной функции - "Alert"

Аргумент функции:

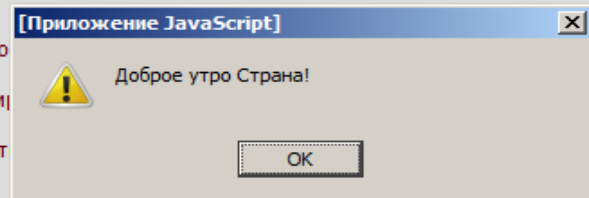
1. строка символов "..." или имя переменной - текст

Функция выводит на экран окно. Фокус ввода сконцентрирован на кнопке "OK".

Аргумент 1 - информационное сообщение для пользователя

Окно имеет элементы управления:

- кнопку "OK"



**Пример 6. «Використання функції Confirm».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript: Приклад використання функції Confirm</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Організація діалогу за допомогою функції Confirm");</SCRIPT>

<P>Сторінка приклад сценарію реалізації діалогу
<P>Діалог організований за допомогою убудованої функції - "Confirm"
<P>Аргумент функції:
<OL>
<LI>рядок символів "...". або ім'я змінної - текстове повідомлення
</OL>
<P>Функція виводить на екран вікно. Фокус введення сконцентрований на цьому вікні.
<P>Аргумент 1 - інформаційне повідомлення для користувача
<P>Вікно має елементи керування:
<UL>
<LI>кнопку "Ok"
<LI>кнопку "Cancel"
</UL>
<SCRIPT language=JavaScript type=text/javascript>
confirm("Хочеш солодких апельсинів?");
</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує проміжний результат, крок 1:

**Лабораторный практикум по языку JavaScript**

**Пример: Организация диалога при помощи функции Confirm**

Страница пример сценария реализации диалога

Диалог организован при помощи встроенной функции - "Confirm"

Аргумент функции:

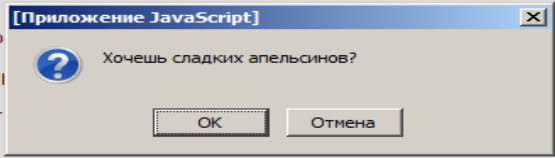
1. строка символов "..." или имя переменной - текст

Функция выводит на экран окно. Фокус ввода сконцентрирован на этом окне.

Аргумент 1 - информационное сообщение для пользователя

Окно имеет элементы управления:

- кнопку "Ok"
- кнопку "Cancel"



**Пример 7. «Простий рахунок (Арифметика)».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Практикум JavaScript. Виконання найпростіших
арифметичних операцій</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Виконання найпростіших арифметичних операцій");</SCRIPT>
<P>Сторінка включає приклад коду JavaScript виконання арифметичних операцій.
<P>Реалізована просте завдання перерахування значення температури по
Цельсию
у значення по Кельвіні, а потім у значення по Фаренгейту.
<P>Запит вихідних даних виконується в діалозі.
<P>Результати обчислень виведені в інформаційному вікні й у документ не
записані</P>
<P>Текст сценарію включає наступні інструкції:
<OL>
<LI>var x,y; // оголошення змінних
<LI>x=prompt("Уведіть температуру повітря за Цельсієм","0"); // запит значення
<LI>x=parseFloat(x); // переклад інформації з тексту в
число
<LI>y=273+x; // обчислення (оператор присвоювання)
<LI>alert("Температуру повітря за Кельвінім : "+y); // вивід інформаційного вікна
<LI>y=x*5/7; // обчислення (оператор присвоювання)
<LI>alert("Температуру повітря за Фаренгейтом : "+y); // вивід інформаційного
вікна
</OL>
<SCRIPT language=JavaScript type=text/javascript> // початок коду сценарію мовою
JavaScript

    var x,y; // оголошення змінних
    x=prompt("Уведіть температуру повітря за Цельсієм","0"); // запит значення
    x=parseFloat(x); // переклад інформації з тексту в число
    y=273+x; // обчислення (оператор присвоювання)
    alert("Температуру повітря за Кельвінім : "+y); // вивід інформаційного вікна
    y=x*5/7; // обчислення (оператор присвоювання)
    alert("Температуру повітря за Фаренгейтом : "+y); // вивід інформаційного вікна

</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```



## Формує проміжний результат, крок 1:

**Лабораторный практикум по языку JavaScript**

**Пример: Выполнение простейших арифметических операций.**

Страница содержит пример кода JavaScript выполнения арифметических операций.

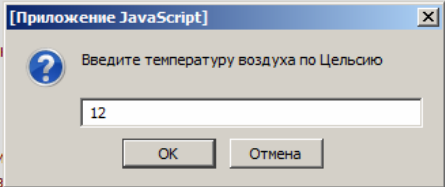
Реализована простая задача пересчета значения температуры по Цельсию в значение по Кельвину, а затем в значение по Фаренгейту.

Запрос исходных данных выполняется в диалоге.

Результаты вычислений выведены в информационном окне.

Текст сценария содержит следующие инструкции:

1. var x,y; // объявление переменных
2. x=prompt("Введите температуру воздуха по Цельсию");
3. x=parseFloat(x); // перевод информации из текста в число
4. y=273+x; // вычисление (оператор присваивания)
5. alert("Температуру воздуха по Кельвину : "+y); // вывод информационного окна
6. y=x\*5/7; // вычисление (оператор присваивания)
7. alert("Температуру воздуха по Фаренгейту : "+y); // вывод информационного окна



## Формує проміжний результат, крок 2:

**Лабораторный практикум по языку JavaScript**

**Пример: Выполнение простейших арифметических операций.**

Страница содержит пример кода JavaScript выполнения арифметических операций.

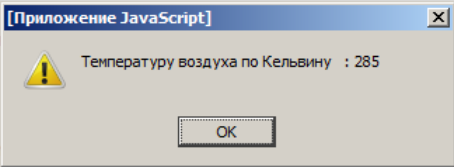
Реализована простая задача пересчета значения температуры по Цельсию в значение по Кельвину, а затем в значение по Фаренгейту.

Запрос исходных данных выполняется в диалоге.

Результаты вычислений выведены в информационном окне.

Текст сценария содержит следующие инструкции:

1. var x,y; // объявление переменных
2. x=prompt("Введите температуру воздуха по Цельсию");
3. x=parseFloat(x); // перевод информации из текста в число
4. y=273+x; // вычисление (оператор присваивания)
5. alert("Температуру воздуха по Кельвину : "+y); // вывод информационного окна
6. y=x\*5/7; // вычисление (оператор присваивания)
7. alert("Температуру воздуха по Фаренгейту : "+y); // вывод информационного окна



## Формує проміжний результат, крок 2:

**Лабораторный практикум по языку JavaScript**

**Пример: Выполнение простейших арифметических операций.**

Страница содержит пример кода JavaScript выполнения арифметических операций.

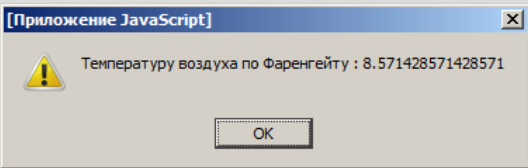
Реализована простая задача пересчета значения температуры по Цельсию в значение по Кельвину, а затем в значение по Фаренгейту.

Запрос исходных данных выполняется в диалоге.

Результаты вычислений выведены в информационном окне.

Текст сценария содержит следующие инструкции:

1. var x,y; // объявление переменных
2. x=prompt("Введите температуру воздуха по Цельсию");
3. x=parseFloat(x); // перевод информации из текста в число
4. y=273+x; // вычисление (оператор присваивания)
5. alert("Температуру воздуха по Кельвину : "+y); // вывод информационного окна
6. y=x\*5/7; // вычисление (оператор присваивания)
7. alert("Температуру воздуха по Фаренгейту : "+y); // вывод информационного окна



**Пример 8. А от і перший діалог "Що ти хочеш - Одержуй!"**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад: Динамічна побудова таблиці</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Динамічна побудова таблиці");</SCRIPT>
<SCRIPT JavaScript type=text/javascript>
    var i, j, minrow, maxrow, mincol, maxcol;
    alert("Зараз буде динамічно побудована таблиця множення.");
    minrow=prompt("Уведіть мінімальне значення для множника 1","1")
    maxrow=prompt("Уведіть максимальне значення для множника 1","1")
    mincol=prompt("Уведіть мінімальне значення для множника 2","1")
    maxcol=prompt("Уведіть максимальне значення для множника 2","1")
    document.write("<TABLE BORDER=4 CELSPACING=5 CELLPADDING=5 Rules=cols")
    document.write("<Caption>Таблиця множення</Caption>");
    document.write("<TR><TH></TH>")
    for (j=mincol; j<=maxcol; j++) document.write("<TH>"+j+"</TH>");
    document.write("</TR>");

    for (i=minrow; i<=maxrow; i++)
    {
        document.write("<TR>")
        document.write("<TD><STRONG>"+i+"</STRONG></TD>");
        for (j=mincol; j<=maxcol; j++)
            document.write("<TD>"+(i*j)+"</TD>");
        document.write("</TR>");
    }
    document.write("</TABLE>");
</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

Таблиця умножень		
4	5	6

**Пример 9. «Функції: корінь квадратний і зведення в ступінь».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Об'єкт Math(Математика), приклад обчислення
квадратного кореня числа й функція зведення в ступінь.</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Обчислення квадратного кореня числа й функція зведення в ступінь
(об'єкт Math(Математика)).");</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>

var thenumber;
var mistake=3;
var allmistake=0;
do          // Забезпечимо правильне введення тільки числових значень
{
    var ask=1; // тимчасова змінна, значення 1 будемо вважати
               // що введення буде виконаний правильно
    thenumber=prompt("Для обчислення кореня числа введіть його","0");// а тут діалог -
очікуємо від користувача значення

    if(isNaN(thenumber)) // перевіримо "уведений рядок є числом"
    {
        alert("А ввели не число! Повторите введення"); // Повідомимо клієнтові
        ask=0;           // Відзначимо: введення виконане невірно
        allmistake++;     // Уважаємо кількість помилок введення
    }
}while( (!ask) && (mistake !=allmistake) );

if(mistake==allmistake)
document.writeln("<h3>Ви допустили помилки при уведенні даних 3 рази. Почніть
спочатку!!!</h3>");
else
{
thenumber=parseFloat(thenumber);// перетворимо текст у речовинне число
kor=Math.sqrt(thenumber); // обчислимо квадратний корінь
korint=Math.floor(kor);    // округлимо до найближчого цілого ВНИЗ
korintup=Math.ceil(kor);   // округлимо до найближчого цілого НАГОРУ
document.write("<h3>Квадратний корінь числа "+thenumber+" дорівнює "+kor+"</h3>");
document.write(kor+"<sup>+2+</sup>+"="+thenumber+"<br>");
document.write(korint+"<sup>+2+</sup>="+Math.pow(korint,2)+"<br>");
document.write(korintup+"<sup>+2+</sup>="+Math.pow(korintup,2)+"<br>");
}
</SCRIPT>

```

```
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>  
</BODY>  
</HTML>
```



Формує результат:

**Лабораторный практикум по языку JavaScript**

Пример: Вычисление квадратного корня числа и функция возведения в степень (объект Math(Математика)).

**Квадратный корень числа 5 равен 2.23606797749979**

$$2.23606797749979^2=5$$
$$2^2=4$$
$$3^2=9$$

Последнее обновление документа: 10/25/2010 19:02:05  
Харьковская национальная академия городского хозяйства  
Кафедра Прикладной математики и Информационных технологий  
телефоны:  (+38 057) 707-31-45 ,  (+38 057) 707-31-31 )  
Жду пожеланий и замечаний, адрес для корреспонденций:  
[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)  
[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)  
С уважением, Яковичский Игорь Леонидович

**Пример 10. «Формування таблиці квадратів».**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад циклічних обчислень: "Формування таблиці
квадратів".</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Формування таблиці квадратів"); // функція з особистої бібліотеки
var i, j, jn=1, jk=10;
document.write("<TABLE BORDER=2 CELSPACING=2 CELLPADDING=2 RULES=cols>");
for (i=1; i<=20; i++)
{
document.write("<TR>");
for (j=jn; j<=jk; j++)
{
document.write("<TD>"+j+"="+j*j+"</TD>");
}
document.write("</TR>");
jn+=10;
jk+=10;
}
document.write("</TABLE>");

jac_show(); // функція з особистої бібліотеки
</SCRIPT>

</BODY>
</HTML>
```

**Пример 11. Таблиця множення**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад: Таблиця множення</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Формування таблиці множення");
var i, j;
document.write("<TABLE BORDER=4 CELSPACING=5 CELLPADDING=5 RULES=cols>");
    for (i=1; i<=10; i++)
    {
        document.write("<TR>");

        for (j=1; j<=10; j++)
            document.write("<TD>"+j+"&times;"+i+"="+i*j+"</TD>");

        document.write("</TR>");
    }
    document.write("</TABLE>");
jac_show();
</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

Пример: Формирование таблицы умножения									
1×1=1	2×1=2	3×1=3	4×1=4	5×1=5	6×1=6	7×1=7	8×1=8	9×1=9	10×1=10
1×2=2	2×2=4	3×2=6	4×2=8	5×2=10	6×2=12	7×2=14	8×2=16	9×2=18	10×2=20
1×3=3	2×3=6	3×3=9	4×3=12	5×3=15	6×3=18	7×3=21	8×3=24	9×3=27	10×3=30
1×4=4	2×4=8	3×4=12	4×4=16	5×4=20	6×4=24	7×4=28	8×4=32	9×4=36	10×4=40
1×5=5	2×5=10	3×5=15	4×5=20	5×5=25	6×5=30	7×5=35	8×5=40	9×5=45	10×5=50
1×6=6	2×6=12	3×6=18	4×6=24	5×6=30	6×6=36	7×6=42	8×6=48	9×6=54	10×6=60
1×7=7	2×7=14	3×7=21	4×7=28	5×7=35	6×7=42	7×7=49	8×7=56	9×7=63	10×7=70
1×8=8	2×8=16	3×8=24	4×8=32	5×8=40	6×8=48	7×8=56	8×8=64	9×8=72	10×8=80
1×9=9	2×9=18	3×9=27	4×9=36	5×9=45	6×9=54	7×9=63	8×9=72	9×9=81	10×9=90
1×10=10	2×10=20	3×10=30	4×10=40	5×10=50	6×10=60	7×10=70	8×10=80	9×10=90	10×10=100

**Пример 12. Циклічні обчислення. Факторіал цілого числа N.****Лабораторный практикум по языку JavaScript****Пример: Циклические вычисления на примере Вычисление факториала**

Задача вычисления факториала числа  $n$  имеет в математике не последнее значение. Факториал  $n$  обозначается в математике  $n!$  и обозначает последовательное умножение целых чисел от 1 до  $n$ . Иными словами  $n! = 1 * 2 * 3 * 4 * \dots * (n-1) * n$ . Ну чем Вам не циклические вычисления.

Для описания работы операторов цикла приведем 3 примера программ. Все они решают одну задачу, разными являются только средства языка JavaScript.

1. оператор **For** (инициализация ; условие ; операция, выполняемая после шага цикла) { тело цикла }
2. оператор **While** (условие) { тело цикла }
3. оператор **Do** { тело цикла } **while** (условие)

**Оператора цикла FOR (повний текст документа HTML).**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Приклад: Обчислення факторіала за допомогою
оператора циклу FOR.</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Обчислення факторіала за допомогою оператора циклу
FOR»);</SCRIPT>

<BR>Уведіть значення n і натисніть клавішу "Tab"<BR>
<INPUT type="text" name="mygetnumber" value="1" onchange="getn();"> <BR>

Факторіал дорівнює<BR>
<TEXTAREA name="fact" rows="1" cols="30"></TEXTAREA>

<SCRIPT language=JavaScript type=text/javascript>
    // Функція обчислення факторіала
function factorial(n)      // оголошення функції з одним формальним параметром
{
    var z=1;                // внутрішня перемінна, у якій буде отриманий результат
    var i;                  // внутрішня перемінна - лічильник циклу
    for( i=1 ; i<=n ; i++ ) // заголовок циклу: почати цикл значенням лічильника 1 і
    виконувати
                                // доти поки
                                // лічильник циклу менше або дорівнює значенню перемінної n
                                // після завершення кожного кроку циклу збільшувати лічильник
на 1
    {
        z*=i;                // розкриваємо тіло циклу
                                // оператор тіла циклу: перемінної z присвоїти результат
                                // виразу z*i
    }                          // закінчення тіла циклу
```

```

return z;           // обчислене значення змінної z повертається з функції
}
//*****
// Функція getn - не має формальних параметрів
// Її виклик пов'язаний з подією "Значення поля змінене" - onchange
// для текстового поля mygetnumber
//*****
function getn()
{
//*****
// document.all.mygetnumber.value - значення текстового поля з ім'ям mygetnumber
// є фактичним значенням формального параметра n функції factorial
// обчислене значення факторіала присвоюємо document.all.fact.value
// значення поля textarea с ім'ям fact
//*****

    document.all.fact.value=factorial(document.all.mygetnumber.value);
}
</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Оператора циклу DO ... WHILE (фрагмент - функція factorial).

```

// Функція обчислення факторіала
function factorial(n) // оголошення функції з одним формальним параметром
{
    var z=1;          // внутрішня перемінна, у якій буде отриманий результат
    var i=0;          // оголошення й ініціалізація внутрішньої перемінної - лічильник циклу
    do                // заголовок циклу: почати цикл
    {
        // розкриваємо тіло циклу
        i++;          // перемінну i збільшуємо на 1
        z*=i;         // перемінної z присвоюємо результат виразу z*i
    } while(i<n);     // поки перемінна i менше значення перемінної n продовжуємо цикл
    return z;         // обчислене значення перемінної z повертаємо з функції
}

```

Оператора циклу WHILE (фрагмент - функція factorial).

```

// Функція обчислення факторіалаfunction
factorial(n)        // оголошення функції з одним формальним параметром
{
    var z=1;          // оголошення й ініціалізація внутрішньої перемінної, у якій буде
отриманий результат
    var i=1;          // оголошення й ініціалізація внутрішньої перемінної - лічильник для
циклу
    while(i<n)        // заголовок циклу: почати цикл і виконувати доти поки перемінна i
буде менше або дорівнює значенню перемінної n
    {
        // розкриваємо тіло циклу
        z*=i;         // перемінної z присвоюємо результат виразу z*i
        i++;          // перемінну i збільшуємо на 1
    }
}

```



```
} // закінчення тіла циклу  
return z; // обчислене значення перемінної z повертаємо з функції  
}
```

Формує результат:

### Лабораторный практикум по языку JavaScript

---

#### Пример: Вычисление факториала с помощью оператора цикла

---





Введите значение n и нажмите клавишу "Tab"

Факториал равен

---

Последнее обновление документа: 10/29/2010 00:13:00  
Харьковская национальная академия городского хозяйства  
Кафедра Прикладной математики и Информационных технологий  
телефоны:  (+38 057) 707-31-45  ,  (+38 057) 707-31-31   
Жду пожеланий и замечаний, адрес для корреспонденций:  
[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)  
[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)  
С уважением, Яковичский Игорь Леонидович

**Пример 13. «Підрахунок кількості символів у тексті».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Практикум JavaScript. Приклад: Підрахунок входження символу(слова) у
тексті.</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Об'єкт String: властивості й методи. Підрахунок входження символу(слова)
у тексті.");</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>
// Функція обчислення кількості входження символу
function count_symbol(str,sym) // оголошення функції із двома формальними
параметрами
{
    var len= str.length; // у перемінній len запам'ятаємо довжину уведеного рядка -
використали властивість об'єкта
    var start=0; // перемінна для номера позиції символу в рядку
    var count=0; // лічильник для результату
    while(start!=-1) // заголовок циклу: і виконувати поки перемінна start не
дорівнює -1
    {
        // відкриваємо тіло циклу
        // метод indexOf об'єкта String - пошук позиції входження підстроки в рядок
        start = str.indexOf(sym,start);
        if(start!=-1)
        {
            count++; // символ знайдений збільшимо лічильник
            start++; // перейдемо до наступної позиції у вихідному рядку
        }
    } // закінчення тіла циклу
    return count; // обчислене значення перемінної count повертаємо з функції
}

function getcount(txt,s)
{
    document.all.symbol_count.value = count_symbol( document.all.put_text.value ,
    document.all.symbol.value );
}
</script>
<br>Уведіть текст<br>
<textarea name="put_text" cols="60" rows="3">Мама, мама, що я буду робити?
Мама, мама, з ким я буду жити?</textarea>
<br>Уведіть символ(слово)<br>
<input type="text" name="symbol" type="text">

<input type="button" name="count_btn" value="Обчислити"
onClick="getcount(document.all.put_text.value,document.all.symbol.value);"><br>

```

```

Буква зустрічається в тексті<br>
<textarea name="symbol_count" cols="10" rows="1"></textarea>раз
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

**Лабораторний практикум по языку JavaScript**

---

Пример: Объект String: свойства и методы. Подсчет вхождения символа(слова) в текст.

---

Введите текст

Мама, мама, что я буду делать? Мама, мама, с кем я буду жить?

Введите символ(слово)

Буква встречается в тексте

раз

---

Последнее обновление документа: 10/29/2010 00:29:22

Харьковская национальная академия городского хозяйства

Кафедра Прикладной математики и Информационных технологий

телефоны: (+38 057) 707-31-45 , (+38 057) 707-31-31

Жду пожеланий и замечаний, адрес для корреспонденций:

[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)

[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)

С уважением, Яковичский Игорь Леонидович

**Пример 14. «Обчислення суми: Перерахуємо грошки в кишені».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум Javascript. Приклад циклічних обчислень: Перерахуємо наші
грошки</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Циклічні обчислення. Перерахуємо наші грошки");</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript> // Починається текст програми(скрипта)
var denga;    // перемінна для введення номіналу монети або купюри
var coshel=0; // перемінна для нагромадження значення суми
              // перемінна здобула значення 0
              // тобто ініціалізована початковим значенням перед використанням
alert("Почнемо рахувати гроші в кишені!");    // просто балаканина

do            // Починаємо циклічні обчислення
{            // Розкриваємо тіло циклу
            // а тут діалог - очікуємо від користувача значення
denga=prompt("Уведіть номінал монети або купюри в копійках","0");
denga=parseInt(denga);    // перекладемо текст у число
coshe=coshel+denga;        // безпосередньо накопичуємо суму

}            // Закривається тіло циклу
while(denga!=0)            // умова закінчення циклу
                        // цикл закінчується коли умова приймає значення НЕПРАВДА
document.write("<h3>Перерахуємо наші грошки</h3>");
document.write("А в кишені у Вас "+coshe+" копійок<br>");    // вивід результату
document.write("Або "+(coshe/100)+" гривень<br>");            // вивід результату в іншій
форми
</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

**Пример 15. «Нехай він сам не свій, але ми однаково перерахуємо його грошки».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум Javascript. Приклад циклічних обчислень: Перерахуємо наші
грошки (доповнення).</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>
example_show("Циклічні обчислення. Перерахуємо наші грошки (доповнення -
контроль коректності даних)»).</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript> // Починається текст програми(скрипу)
var denga; // перемінна для введення номіналу монети або купюри
var coshel=0; // перемінна для нагромадження значення суми
// перемінна здобула значення 0
// тобто ініціалізована початковим значенням перед використанням
alert("Почнемо рахувати гроші в кишені!"); // просто балаканина
do // Починаємо цикл обчислення
{
    do // Забезпечимо правильне введення тільки числових значень
    {
        var ask=1; // тимчасова змінна, значення 1 будемо вважати
        // що введення буде виконаний правильно
        // а тут діалог - очікуємо від користувача значення
        denga=prompt("Уведіть номінал монети або купюри в копійках","0");
        if(isNaN(denga)) // перевіримо "уведений рядок є числом"
        {
            // Повідомимо клієнтові
            alert("А Ви точно дістали гроші? Ви у своєму розумі?! Введіть ще раз»).);
            ask=0; // помітимо, що введення виконане невірно
        }
    } while(!ask);
    denga=parseInt(denga); // перетворимо рядок у число
    coshel=coshel+denga; // накопичуємо суму
    // вивід інформації на кожному кроці циклу
    document.write ("Ви вже дістали "+coshel+" копійок, або "+(coshel/100)+"
гривень<br>");
} while(denga!=0) // умова закінчення циклу
// цикл закінчується, коли умова приймає значення НЕПРАВДА
document.write ("<hr><h3>Перерахуємо Ваші грошки!</h3>");
// вивід остаточного результату
document.write ("Усього у Вас у кишені "+coshel+" копійок, або "+(coshel/100)+"
гривень<hr>");

</SCRIPT>
<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>

```

```
</BODY>  
</HTML>
```

Формує результат:

### Лабораторный практикум по языку JavaScript





**Пример: Циклические вычисления. Посчитаем наши денежки (дополнение - контроль корректности данных).**

Вы уже достали 10 копеек, или 0.1 гривен  
Вы уже достали 60 копеек, или 0.6 гривен  
Вы уже достали 180 копеек, или 1.8 гривен  
Вы уже достали 180 копеек, или 1.8 гривен

#### Посчитаем Ваши денежки!

Всего у Вас в кармане 180 копеек, или 1.8 гривен

#### Это способ программирования циклических операций при помощи конструкции do ... while

Последнее обновление документа: 10/25/2010 17:40:22  
Харьковская национальная академия городского хозяйства  
Кафедра Прикладной математики и Информационных технологий  
телефоны:  (+38 057) 707-31-45  ,  (+38 057) 707-31-31   
Жду пожеланий и замечаний, адрес для корреспонденций:  
[jac@ksame.kharkov.ua](mailto:jac@ksame.kharkov.ua)  
[igor.yakovitskiy@gmail.com](mailto:igor.yakovitskiy@gmail.com)  
С уважением, Яковицкий Игорь Леонидович

**Пример 16. «Функціональні кнопки. Власна функція».**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<Title>Практикум JavaScript. Приклад: Оголошення й використання(виклик) власної
функції</Title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language=JavaScript type=text/javascript>example_show("Оголошення й
використання(виклик) власної функції");</SCRIPT>

<SCRIPT language=JavaScript type=text/javascript>

var x="birthday_test.htm";    // оголошення й ініціалізація перемінних
var y="image_test.htm";      // x,y

function ShowPage(thepage)    // ShowPage - власне ім'я функції
{                               // thepage - ім'я формального параметра
    window.open(thepage);     // виклик методу Open(розкрити) для об'єкта
    Window(вікно)
}
</script>

<button onClick="ShowPage(x)">День народження</button> <br>
<button onClick="ShowPage(y)">Бендер Остап Ібрагімович</button>

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</body>
</html>

```

**Пример 17. Робота з об'єктом Data.**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>Практикум JavaScript. Робота з об'єктом Data.</TITLE>
<META http-equiv=Content-Type content="text/html; charset=utf-8">
<LINK href="java_css.css" type=text/css rel=stylesheet>
<SCRIPT type="text/javascript" src="jac_library.js"></SCRIPT>
</HEAD>

<BODY>
<SCRIPT language=JavaScript type=text/javascript>example_show("Робота з об'єктом
Data».);</SCRIPT>
<script language="JavaScript">

var thedate = new Date(); // Створення екземпляра об'єкта "Дата" за допомогою
конструктора
                        // атрибуту об'єкта ініціалізує значенням "системний час"
                        // Функції dayWeek, HelloStr - визначені в бібліотеці "jac_library.js"

var show;
show="Сьогодні: ";
show+=thedate.toLocaleDateString(); // звертання до методу toLocaleDateString
show+=" ";
show+=dayWeek(thedate.getDay()); // звертання до методу getDay - отримати номер
дня тижня
show+="<BR>";
document.write(show);
show="Поточний час: ";
show+=thedate.toLocaleTimeString(); // звертання до методу toLocaleTimeString
show+="<BR>";

document.write(show);

show=HelloStr(thedate.getHours()); // звертання до методу getHours - отримати значення
ГОДИНИ
show+="<BR>";

document.writeln(show);

</script>

<SCRIPT language=JavaScript type=text/javascript>jac_show();</SCRIPT>
</BODY>
</HTML>

```

Формує результат:

Сегодня: 30 Жовтень 2010 р., Суббота  
Текущее время: 23:46:50  
Добрый вечер!



## Індивідуальні завдання.

Виконання індивідуальних завдань дозволить розвинути практичні навички написання програм.

### Завдання 1. «Анкета слухача».

Створити документ HTML, що включає код сценарію мовою JavaScript.

Сценарій повинен виконати наступні дії й операції:

- сформувати анкету слухача (студента). Застосувати **методи *write, writeln* об'єкта *document***

Анкету сформувати построчно й включити в неї наступну інформацію:

- Найменування навчального закладу
- Факультет
- Спеціальність / Група
- Прізвище Ім'я По батькові
- Дата народження
- Email

Параметри документа HTML:

- заголовок документа "Завдання 1. ПІБ, Дата формування».
- використати теги заголовка не більше 4-го рівня
- кольори тла довільний
- кольори тексту довільний
- вирівнювання тексту по центрі
- анкету підкреслити горизонтальною лінією на початку й наприкінці

### Завдання 2. «Оформлення власної функції».

Створити документ HTML, що включає код сценарію мовою JavaScript.

Сценарій повинен виконати наступні дії й операції:

- Доповнити код сценарію JavaScript "Анкета слухача" (сформована в завданні 1) рядком: Робота № .
- Сценарій "Анкета слухача" оформити у вигляді власної функції.
- Зберегти сценарій в окремому файлі \*.js, прообраз особистої бібліотеки функцій
- Підключити бібліотеку \*.js до документа HTML
- Викликати функцію виводу анкети слухача
- Вивести номер роботи

---

**Всі наступні завдання варто реалізувати з використання функції із завдання №2**

---

*Завдання 3. «Проста логіка».*

Створити документ HTML, що включає код сценарію мовою JavaScript.

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача
- Вивести номер роботи
- У діалоговому режимі послідовно запросити введення 2-х чисел
- Вивести їх у документ
- Зрівняти числа між собою
- Вивести в документ одну із фраз:
  - число 1 більше числа 2
  - число 1 менше числа 2
  - число 1 дорівнює числу 2

*Завдання 4. «Обчислювальний процес, що розгалужується».*

Створити документ HTML, що включає код сценарію мовою JavaScript.

Документ повинен включати наступні елементи HTML:

- заголовок документа "Завдання 4. ПІБ, Дата формування".
- 6 текстових полів "Координати вершин трикутника"
- текстове поле "Довжини сторін АВ, ВР, СА"
- текстове поле "Тип трикутника"
- текстове поле "Площа трикутника"
- текстове поле "Положення початку координат"

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача
- Вивести номер роботи
- У діалоговому режимі послідовно запросити введення 6-х чисел (x1, y1, x2, y2, x3, y3) - координати вершин трикутника:
- Обчислити довжини сторін трикутника
- Визначити, чи можна побудувати трикутник по цих вершинах
- Якщо трикутник існує, обчислити його площу
- Якщо трикутник існує, визначити його тип:

- Гострокутний
  - Тупокутний
  - Прямокутний
- Якщо трикутник існує, визначити, де розташована крапка початку координат - крапка (0,0)
    - Крапка лежить поза областю трикутника
    - Крапка лежить усередині області трикутника
    - Крапка лежить на стороні трикутника
  - Вивести в поля координат введені числа (x1, y1, x2, y2, x3, y3)
  - Вивести в поле "Довжини сторін АВ, ВР, СА" значення довжин сторін
  - Вивести в поле "Тип трикутника" тип трикутника, якщо він існує, інакше фразу "трикутник не існує"
  - Вивести в поле "Площа трикутника" значення площі, якщо він існує, інакше фразу "трикутник не існує"
  - Вивести інформацію в поле "Положення початку координат"

Для перевірки роботи сценарію використати такі дані:

- крапка А із координатами  $x_1=-2, y_1=1$ ,
- крапка В із координатами  $x_2=2, y_2=2$ ,
- крапка С із координатами  $x_3=1, y_3=-2$ ,

*Завдання 5. «Від числа до тексту й назад».*

Створити документ HTML, що включає код сценарію мовою JavaScript.

Обчислити, скільки чисел з парною кількістю цифр (наприклад, двозначні, чотиризначні й т.д.) мають однакову суму першої й другої половини цифр.

Наприклад: чотиризначні числа 1001, 1010 - є числами, у яких сума перших двох цифр - 2 дорівнює сумі другої пари цифр.

Склад HTML сторінки:

- заголовок документа "Завдання 5. ПІБ, Дата формування".;
- текстове поле "Розрядність аналізованих чисел";
- текстове поле "Початкове значення інтервалу";
- текстове поле "Кінцеве значення інтервалу";
- текстове поле "Кількість чисел";
- поле TEXTAREA "Числа задовольняючій умові".
- кнопку "Обчислити".

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача

- Вивести номер роботи
- Обчислити кількість чисел задовольняючій умові й вивести їх у поле "Результат":
- Вивести їх у поле "Числа задовольняючій умові" перші 10 чисел, числа розділити комі:
- Сценарій починає працювати при натисканні кнопки "Обчислити".

*Завдання 6. «Циклічні обчислення. Табулювання функцій».*

Створити HTML сторінку, що включає код сценарію мовою JavaScript.

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача
- Вивести номер роботи
- Сформувати таблицю математичних функцій у діапазоні від початкового значення аргументу до кінцевого значення аргументу із заданим кроком збільшення аргументу.
- Значення функцій округлити до вказаного значення.

Таблиця повинна мати такий вигляд:

Таблиця математичних функцій					
x	sin(x)	cos(x)	tg(x)	ctg(x)	exp(x)

**Варіанти виконання завдання**

- Межі табулювання й крок табулювання кількість знаків округлення користувач вводить за допомогою полів введення. Початок роботи сценарію починається після натискання "кнопки" "Побудувати таблицю"
- Межі табулювання й крок табулювання кількість знаків округлення сценарій запитує в діалоговому режимі.

*Завдання 7. «Циклічні обчислення. Сортування та інше».*

Створити документ HTML, що включає код сценарію мовою JavaScript.

Склад HTML сторінки:

- заголовок документа "Завдання 7. ПІБ, Дата формування".
- поле TEXTAREA "Вихідні дані" для введення вихідних даних - набір цілих чисел.
- поле TEXTAREA "Дані відсортовані за зростанням" для виводу відсортованого набору цілих чисел.
- поле TEXTAREA "Дані відсортовані за спаданням" для виводу відсортованого набору цілих чисел.

- текстове поле "Максимальне значення" для виводу максимального значення набору.
- текстове поле "Мінімальне значення" для виводу мінімального значення набору.
- текстове поле "Сума елементів" для виводу значення суми елементів.
- текстове поле "Середнє значення" для виводу середнього значення набору елементів.
- кнопку "Обчислити".

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача
- Вивести номер роботи
- Підрахувати кількість уведених чисел набору
- Відсортувати набір чисел у порядку зростання й вивести їх у поле TEXTAREA
- Відсортувати набір чисел у порядку убутання й вивести їх у поле TEXTAREA
- Визначити мінімальний елемент і вивести в текстове поле
- Визначити максимальний елемент і вивести в текстове поле
- Обчислити суму елементів і вивести в текстове поле
- Обчислити середнє значення набору й вивести в текстове поле

#### Завдання 8. «Об'єктне програмування. Об'єкт String».

Створити документ HTML, що включає код сценарію мовою JavaScript.

Склад HTML сторінки:

- заголовок документа "Завдання 8. ПІБ, Дата формування».
- поле TEXTAREA "Текст 1" для введення однієї пропозиції (частини тексту).
- поле TEXTAREA "Текст 2" для введення другої пропозиції (частини тексту).
- поле TEXTAREA "Результат" для виводу з'єднаного тексту.
- текстове поле "Кількість символів у тексті" для виводу кількості символів результуючого тексту.
- текстове поле "Кількість слів" для виводу кількості слів у тексті результуючого тексту.
- текстове поле "Кількість розділових знаків" для виводу кількості розділових знаків результуючого тексту.
- кнопку "Виконати".

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача

- Вивести номер роботи
- об'єднати два вихідні рядки в одну й вивести в поле TEXTAREA
- обчислити й вивести в текстове поле:
  - загальна кількість символів;
  - кількість слів (словом є нерозривна послідовність букв, відділена від іншої послідовності символом «пробіл» або розділовими знаками) і вивести в текстове поле;
  - кількість розділових знаків

Реалізувати операції за допомогою об'єкта **String** його методів.

Для перевірки роботи сценарію використати такі рядки:

- 1 - "Мова програмування JavaScript - ";
- 2 - "потужний засіб програмування в мережі Інтернет».

#### *Завдання 9. «Об'єктне програмування. Масиви - об'єкт Array».*

Створити документ HTML, що включає код сценарію мовою JavaScript.

Склад HTML сторінки:

- заголовок документа "Завдання 9. ПІБ, Дата формування».
- поле TEXTAREA "Речення" для введення речення (послідовність слів з рядкових українських літерів; між сусідніми словами кома, за останнім словом - крапка.).
- текстове поле "Всі голосні літери, які входять у кожне слово";
- текстове поле "Всі приголосні літери, які не входять у жодне слово";
- текстове поле "Всі дзвінкі приголосні літери, які входять хоча б в одне слово";
- текстове поле "Всі глухі приголосні, які не входять хоча б в одне слово";
- текстове поле "Всі приголосні літери, які входять тільки в одне слово";
- текстове поле "Всі глухі приголосні літери, які не входять тільки в одне слово"
- кнопку "Проаналізувати речення".

Сценарій повинен виконати наступні дії й операції:

- Вивести анкету слухача
- Вивести номер роботи
- Провести аналіз речення й заповнити всі поля

Реалізувати операції за допомогою об'єкта **Array** його методів.

Для перевірки роботи сценарію використати речення:

- "мова програмування потужний засіб реалізації алгоритмічної обробки інформації, його знання структурує нашу поведінку».

*Завдання 10. «Лабіринт із мінотавром»*

Створити документ HTML, що включає код сценарію мовою JavaScript. Інтерфейс сторінки HTML довільний. Сценарій повинен реалізувати рішення наступного завдання:

Лабіринт має  $N$  кімнат. Кімнати з'єднані коридорами. Таблиця з'єднань включає пари, які з'єднані коридором. Побудувати шлях переміщення з кімнати з номером  $i$  у кімнату з номером  $j$ .

*Завдання 11.*

Створити документ HTML, що включає код сценарію мовою JavaScript. Інтерфейс сторінки HTML довільний. Сценарій повинен реалізувати рішення наступного завдання:

У даній послідовності  $N$  дробних чисел вибрати зростаючу послідовність найбільшої довжини.

## Задачі підвищеної складності.

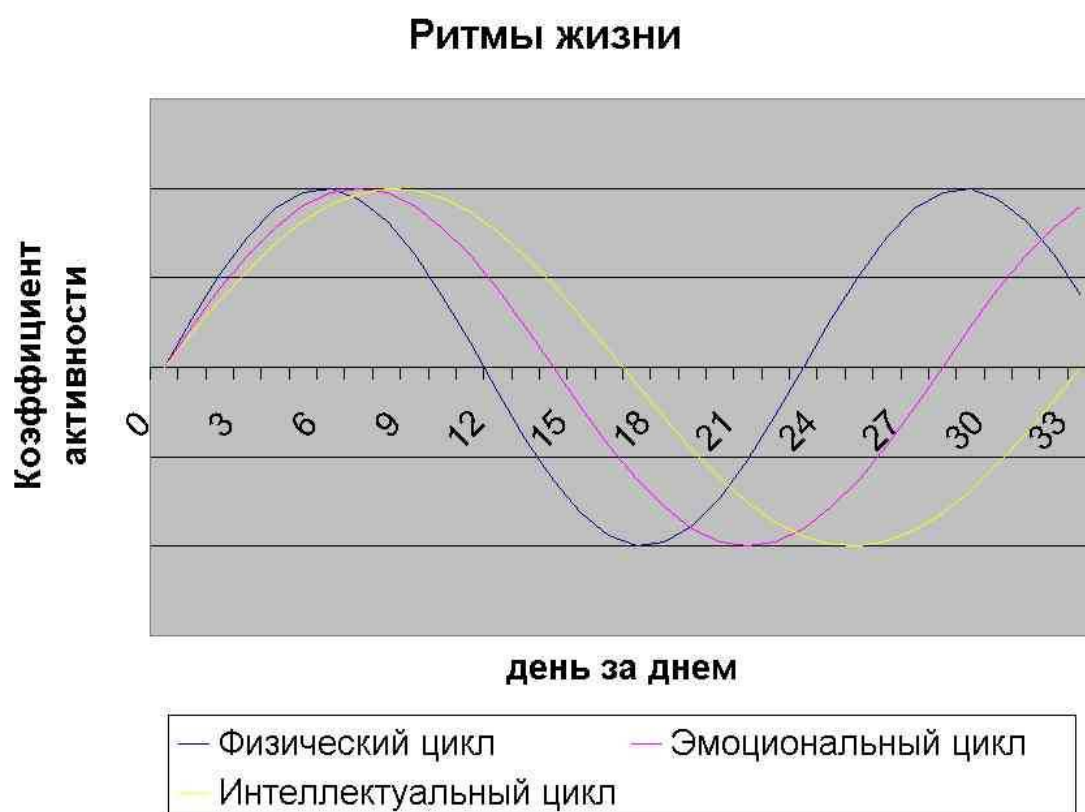
Вирішення задач підвищеної складності поглиблює навички алгоритмічної обробки інформації.

### Завдання «Ритми людського життя».

**Вчені стверджують: життя людини супроводжують періодичні процеси:**

- фізичний - період 23 дня;
- емоційний - період 28 днів;
- психологічний - період 33 дня.

Всі процеси починають відлік у день народження з позитивного півперіоду (аналогічно графіку синусоїди). Вісь X - лінія часу, вважається «лінією життя».



#### Цікаво:

Крапки перетинання циклів з "лінії життя" визначають спеціальні дати в житті людини. Якщо в один день відбувається накладення 3-х циклів, то дата може стати поворотною в житті людини.

Можливо, ми можемо пророчити майбутнє?



**Завдання.**

Сформувати документ HTML, що включає сценарій JavaScript:

- Заголовок сторінки: ПІБ\_виконавця. Лінія життя й події на її шляху.
- Зміст сторінки:

- Прогноз майбутнього: Прізвище Ім'я По батькові
- Дата народження:
- Знак зодіаку:
- Знак за східним календарем:
- Дата початку прогнозу:
- Дата закінчення прогнозу:

Дата	Фізичний цикл	Емоційний цикл	Психологічний цикл
12.12.2011	-	+	+
13.12.2011	*	+	+
14.12.2011	+	+	*
15.12.2011	+	+	-

де:

- «+» відповідає позитивному півперіоду, процес протікає вище "лінії життя";
- «-» відповідає негативному півперіоду, процес протікає нижче "лінії життя";
- «\*» відповідає крапці перетинання циклу з "лінією життя".

Правила колірної оформлення рядків таблиці:

	Цикл	Кольори
1	Фізичний	зелений
2	Емоційний	рожевий
3	Психологічний	блакитний
4	фізичний/емоційний	жовтий
5	фізичний/психологічний	коричневий
6	емоційний/психологічний	жовтогарячий
7	фізичний/емоційний/психологічний	червоний

Підсумкову таблицю заповніть за допомогою сценарію JavaScript. Вихідні дані для роботи сценарію користувач повинен вводити за допомогою полів введення:

- Прізвище Ім'я По батькові;
- Дата народження;
- Дата початку періоду для перегляду;
- Дата закінчення періоду для перегляду.
- 
- 
-

## Завдання «Гра хрестики-нулики».

### Умови гри.

- суперники: Програма - Користувач.
- ігрове поле: - таблиця розміром 3x3 клітки.
- розмір клітки: довільний.
- Клітка таблиці може містити:
- порожнечу
- графічне зображення хрестика
- графічне зображення нулика

### Гра.

- До початку гри програма пропонує супернику:
  - вибрати вид власної фігури для гри (хрестик або нулик).
  - право першого ходу (Ви, Я)
- Гра починається натисканням на кнопку «Початок гри».
- Суперники (програма й користувач) виконують черговий хід по черзі.

### Умови закінчення гри.

- один із суперників установив 3 фігури так, що їх з'єднає пряма лінія;
- користувач натискає на клавіатурі клавішу «Esc»

### Після закінчення гри програма видає одне з повідомлень:

- «Ви перемогли, поздоровляю!»
- «Вибачите, я сама не знаю, як це відбулося! Ви програли».
- «Нічия. Ви прекрасно граєте, але мені сьогодні просто везе. Спасибі за компанію!»

### Вимоги до програми.

- Обчислення по програмі виконуються з HTML-документа.
- Заголовок документа включає фразу "ПІБ слухача». Хрестики-нулики». Дата"
- У тіло документа помістити:
  - зображення поля для гри;
  - поле зі списком для вибору фігури для користувача (Хрестик, Нулик);
  - поле зі списком для вибору права першого ходу (Ви, Я);
  - поле зі списком для вибору швидкості подорожі, значення швидкості "повільно", "швидко", "миттєво";
  - кнопку «Граємо»;
  - кнопку «Закінчити»;

- поле для виводу повідомлення про результати гри (повідомлення описані вище)
- Користувач вибирає фігуру для гри
- Користувач вибирає право першого ходу
- Програма починає працювати після натискання на кнопку «Граємо»
- Програма контролює виконання неправильного ходу - «хід у зайняту клітку».

### **Завдання «Подорож коня по шахівниці».**

Шахівниця - це таблиця 8x8 з клітками довільного розміру, кольори кліток довільний, за умови - клітки однакових кольорів не мають спільних сторін. Клітки адресують по горизонталі літерами латинського алфавіту "A B C D E F G H" по вертикалі арабськими цифрами "1 2 3 4 5 6 7 8". Ім'я клітка визначають як перетинання позначення по горизонталі й по вертикалі, наприклад «A2», «F6». У лівому нижньому куті шахівниці розташована клітка з ім'ям «A1».

Кінь хитра шахова фігура. Кінь виконує один хід за наступними правилами – зсув по горизонталі (*вертикалі*) на 2(1) клітки + зсув по вертикалі (*горизонталі*) на 1(2) клітку. Послідовність напрямків довільна.

#### **Формулювання завдання.**

Установити на шахівниці коня в довільну клітку й послідовно обійти клітки шахівниці, так щоб у кожній побувати тільки один раз.

#### **Вимоги до програми.**

- Обчислення по програмі виконуються з HTML-документа.
- Заголовок документа включає фразу "ПІБ слухача. «Подорож коня по шахівниці». Дата"
- У тіло документа помістити:
  - зображення шахівниці;
  - поле зі списком для вибору імені клітки по вертикалі (літер);
  - поле зі списком для вибору імені клітки по горизонталі (цифра);
  - поле зі списком для вибору швидкості подорожі, значення швидкості "повільно", "швидко", "миттєво";
  - кнопку «Початок подорожі»;
  - кнопку «Закінчити»;
  - поле "Варіант"
- Користувач вибирає значення для початкового розташування коня й вибирає швидкість подорожі

- Програма починає працювати після натискання на кнопку «Початок подорожі»
  - встановлює коня в обрану клітку
  - оцінює клітку для переходу
  - переходить у неї
  - відзначає новими кольором й номером кроку
  - повторює переходи доти, поки не кінь не оббіжить всі клітки
  - після повного обходу заповнює поле "Варіант" цілим числом
  - Очікує дій користувача
- Для перегляду наступного варіанта користувач вибирає значення нове значення початкового розташування коня;
- Натискає на кнопку «Початок подорожі»;
- Для завершення роботи програми користувач натискає кнопку «Закінчити»

### *Завдання «8 з ферзів на шахівниці».*

Шахівниця - це таблиця 8x8 з клітками довільного розміру, кольори кліток довільний, за умови - клітки однакових кольорів не мають спільних сторін. Клітки адресують по горизонталі літерами латинського алфавіту "A B C D E F G H" по вертикалі арабськими цифрами "1 2 3 4 5 6 7 8". Ім'я клітка визначають як перетинання позначення по горизонталі й по вертикалі, наприклад «A2», «F6». У лівому нижньому куті шахівниці розташована клітка з ім'ям «A1».

Ферзь найдужча шахова фігура. Ферзь виконує хід із клітки розташування в клітку мети в будь-якому напрямку шахівниці на будь-яку кількість кліток. Всі клітки на шляху ходу Ферзя вважають «вбитими». Будь-яка фігура, що перебуває на шляху руху ферзя може бути вбита.

#### **Формулювання завдання.**

Розмістити на шахівниці 8 шахових фігур "ферзь" так, щоб вони не "убивали" один одного.

#### **Вимоги до програми.**

- Обчислення по програмі виконуються з HTML-документа.
- Заголовок документа включає фразу "ПІБ слухача». 8 з ферзів на шахівниці». Дата"
- У тіло документа помістити:
  - зображення шахівниці;
  - кнопку «Ставай»;

- кнопку "Закінчити";
- поле "Варіант"
- Програма починає працювати після натискання на кнопку «Ставай»
  - Встановлює ферзя в довільну клітку шахівниці
  - Позначає клітки "вбиті" ферзем (змінює кольори клітки)
  - повторює пункти, поки не розташує на шахівниці 8 ферзів
  - після розташування 8-ої ферзі заповнює поле "Варіант" цілим числом
  - Очікує дій користувача
- Для перегляду наступного варіанта користувач може натиснути на кнопку "Ставай"
- Для завершення роботи програми користувач натискає кнопку "Закінчити"

### *Завдання «8 мирних тур на шахівниці».*

Шахівниця - це таблиця 8x8 з клітками довільного розміру, кольори кліток довільний, за умови - клітки однакових кольорів не мають спільних сторін. Клітки адресують по горизонталі літерами латинського алфавіту "A B C D E F G H" по вертикалі арабськими цифрами "1 2 3 4 5 6 7 8". Ім'я клітка визначають як перетинання позначення по горизонталі й по вертикалі, наприклад «A2», «F6». У лівому нижньому куті шахівниці розташована клітка з ім'ям «A1».

Тура сильна шахова фігура. Вона виконує один хід із клітки розташування в клітку мети в горизонтальному й вертикальному напрямку шахівниці на будь-яку кількість кліток. Всі клітки на шляху ходу Тури вважаються «битими», тобто будь-яка фігура, що перебуває на шляху руху Тури може бути вбита нею.

### **Формулювання завдання.**

Розмістити на шахівниці 8 шахових фігур "тура" так, щоб вони не "убивали" один одного.

### **Вимоги до програми.**

- Обчислення по програмі виконуються з HTML-документа.
- Заголовок документа включає фразу "ПІБ слухача». 8 мирних тур на шахівниці». Дата"
- У тіло документа помістити:
  - зображення шахівниці;
  - кнопку «Ставай»;
  - кнопку "Закінчити";
  - поле "Варіант"

- Програма починає працювати після натискання на кнопку «Ставай»
  - Установлює туру в довільну клітку
  - Позначає клітки "биті" турою (змінює кольори клітки)
  - повторює пункти, поки не розташує на шахівниці 8 тур
  - після установки 8-ий тури заповнює поле "Варіант" цілим числом
  - Очікує дій користувача
- Для перегляду наступного варіанта користувач може натиснути на кнопку "Ставай"
- Для завершення роботи програми користувач натискає кнопку "Закінчити"

### *Завдання «Поділимо золото предків чесно!»*

Друзі-золотошукачі знайшли скарб золотих прикрас древньої цивілізації Інків. Вони вирішили поділити їх по вазі. Домовилися, що розділ буде чесним, якщо різниця між двома частинами буде мінімальна. Кому ж дістанеться частина більшої ваги, вирішить "монета".

Як правильно поділити золото?

#### **Математичне формулювання задачі.**

Є  $N$  чисел  $A_1, A_2, \dots, A_n$ . Варто розподілити ці числа на дві групи так, щоб сума чисел груп була така, що їхня різниця мінімальна.

#### **Вимоги до програми.**

- Обчислення по програмі виконуються з HTML-документа.
- Заголовок документа включає фразу "ПІБ слухача». Поділимо золото предків чесно!» Дата"
- У тіло документа помістити:
  - поле "Кількість монет"
  - поле "Загальна вага золота"
  - поле "Більша частина"
  - поле "Менша частина"
  - поле "Різниця у вазі"
  - кнопку "Розділимо золото"
- Робота програми починається після введення в поле "Кількість монет" числа й натискання на кнопку "Розділимо золото".
- Значення для  $N$  чисел (ваги монет) згенерувати випадковим чином (генератор випадкових чисел).
- Вага будь-якої монети коливається в межі від 20 до 30 грам.

- Обчислити загальну вагу всіх предметів і вивести в поле "Загальна вага золота".
- Сформувати таблицю, в клітки якої вивести значення ваги кожної з "монет".
- Розділити монети на дві групи.
- Обчислити вагу кожної групи.
- Вивести результати в поля введення.
- Більше значення вивести в поле "Більша частина".
- Менше значення вивести в поле "Менша частина".
- Обчислити різницю між вагами частин і вивести результат у поле "Різниця у вазі".

## Список літератури

1. Камер Д. Компьютерные сети и Internet. Разработка приложений для Internet. — М.: Вильямс, 2002
2. Гарнаев А. Ю., Матросов А. В., Новиков Ф. А. и др. Microsoft Office 2000: разработка приложений. — СПб.: БХВ-Петербург, 2000
3. Вейтман В. Программирование для Web. — К.: Диалектика, 2000.— 368с.
4. Шапошников И. В. Интернет-программирование. - СПб.: BHV, 2000. — 224с.
5. Шафран Э. Создание Web-страниц. Самоучитель. - СПб.: Питер, 1999.— 320 с.
6. Аллен Р. Вайк, Джейсон Д. Джиллиам JavaScript. Полное руководство. — М.: Вильямс, 2004. - 720 с., с ил.
7. Слепцова Л.Д., Бидасюк Ю.М. JavaScript. Самоучитель. К.Диалектика, 2008. — 448с., с ил.
8. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. СПб.. Питер, 2001
9. Дэнни Гудман, Майкл Моррисон JavaScript. Библия пользователя. К.Диалектика, 2006. — 1184с., с ил.
10. Методичні вказівки до виконання лабораторних робіт "Створення гіпертекстового документу", Журавок О.Г., Яковицький І.Л., ХНАМГ, 2004., 41 с.
11. Сайт Центру дистанційного навчання Харківської національної академії міського господарства, електронна адреса:  
[www.ksame.kharkov.ua/moodle](http://www.ksame.kharkov.ua/moodle)

## НАВЧАЛЬНЕ ВИДАННЯ

**Журавок** Олексій Геннадійович  
**Яковицький** Ігор Леонідович

Лабораторний практикум «JavaScript. Програмування в Інтернет» з навчальної дисципліни «Обчислювальна техніка, алгоритмічні мови та програмне забезпечення» (для студентів 2 курсу денної форми навчання освітньо-кваліфікаційного рівня бакалавр у галузі знань 0507 - «Електротехніка та електромеханіка» за напрямом підготовки 6.050701 – «Електротехніка та електротехнології», професійного спрямування - «Світлотехніка та джерела світла», «Електротехнічні системи електроспоживання»).

*Відповідальний за випуск: І. Л. Яковицький*

*Редактор: З. І. Зайцева*

*Комп'ютерне верстання: І. Л. Яковицький, І. В. Волосожарова*

План 2009, поз. 526М, 527М

---

Підп. до друку 09.11.2010

Друк на ризографі.

Зам. №

Формат 60 x 84 1/16

Ум. друк. арк. 2,0

Тираж 50 пр.

Видавець і виготовлювач:

Харківська національна академія міського господарства,  
вул. Революції, 12, Харків, 61002

Електронна адреса: rectorat@ksame.kharkov.ua

Свідоцтво суб'єкта видавничої справи:

ДК №731 від 19.12.2001